1

A Practical Time-Delay Estimator for Localizing Speech Sources with a Microphone Array

Michael S. Brandstein, John E. Adcock, and Harvey F. Silverman
Laboratory for Engineering Man/Machine Systems
Division of Engineering
Brown University
Providence, RI 02912
email: msb@lems.brown.edu tel. (401) 863-1044
email: jea@lems.brown.edu tel. (401) 863-1044
email: hfs@lems.brown.edu tel. (401) 863-2276
(fax) (401) 863-1157

June 20, 1995

[1]This work partially funded by DARPA/NSF Grant IRI-8901882, and NSF Grants MIP-9120843 and MIP-9314625.

Abstract

A frequency-domain-based delay estimator is described, designed specifically for speech signals in a microphone-array environment. It is shown to be capable of obtaining precision delay estimates over a wide range of SNR conditions and is simple enough computationally to make it practical for real-time systems. A location algorithm based upon the delay estimator is then developed. With this algorithm it is possible to localize talker positions to a region only a few centimeters in diameter (not very different from the size of the source), and to track a moving source. Experimental results using data from a real 16-element array are presented to indicate the true performance of the algorithms.

1 Introduction

A steerable array of microphones has the potential to replace the traditional head-mounted or desk-stand microphone as the input transducer system for acquiring speech data in many applications. An array of microphones has a number of advantages over a single-microphone system. First, it may be electronically aimed to provide a high-quality signal from a desired source location while it simultaneously attenuates interfering talkers and ambient noise. In this regard, an array has the potential to outperform a single, well-aimed, highly-directional microphone. Second, an array system does not necessitate local placement of transducers, will not encumber the talker with a hand-held or head-mounted microphone, and does not require physical movement to alter its direction of reception. These features make it advantageous in settings involving multiple or moving sources. Finally, it has potential capabilities that a single microphone does not; namely automatic detection, location, and tracking of active talkers in its receptive area. Existing array systems have been used in a number of applications. These include teleconferencing (Flanagan 1985; Kellerman 1991), speech recognition (Silverman 1987; Che, Lin, Pearson, deVries, and Flanagan 1994), speaker identification (Lin, Jan, and Flanagan 1994), speech acquisition in an automobile environment (Grenier 1992; Oh, Viswanathan, and Papamichalis 1992), large-room recording-conferencing (Flanagan, Johnson, Zahn, and Elko 1985), and hearing aid devices (Greenberg and Zurek 1992). These systems also have the potential to be beneficial in several other environments, the performing arts and sporting communities, for instance.

A fundamental requirement of any speech-array application is the ability to determine the relative time delay between signal arrivals at distinct microphone locations. The precision and robustness of these estimates is a crucial factor in the quality of an associated source-location scheme. In addition to high accuracy, these delay estimates must be updated frequently in order to be useful in tracking and beamforming applications. Consider the problem of beamforming to a moving source. It has been shown that for sources in close proximity to the microphones, the array aiming location must be accurate to within a few centimeters to prevent high-frequency rolloff in the received signal (Flanagan and Silverman 1992). An effective beamformer must therefore be capable of including a continuous and accurate location procedure within the beamforming algorithm. This requirement necessitates the use of a delay estimator capable of fine resolution at a high update rate. Any such estimator would also have to be computationally non-demanding to make it practical for real-time systems.

In general, correlation strategies have been used for estimating the time delay between signals received at two spatially distinct sensors. Specifically, the cross-correlation function of the two signals is computed, filtered in some "optimal" sense, and the maximum is found with a peak detector (Hassab 1989; Knapp and Carter 1976). While the filtering criteria and the methods used for peak detection vary considerably, these techniques are all based on maximizing the cross-correlation function. Estimate resolution is limited by the sampling period unless some kind of interpolation method is employed. These methods range from upsampling the signal to parabolic fitting of the cross-correlation function (Hassab 1989); for each there is a general trade-off between the increased accuracy achieved and the computational expense incurred by the procedure. This genre of delay estimation has been applied to the same problem addressed in this paper, talker location in the near field of a microphone array. In (Alvarado 1990) a search method was employed to localize the source position through maximizing a cross-correlation over the x-y positions in the room. More recently, a talker location algorithm (Silverman and Kirtman 1992) based on multirate interpolation of the cross-correlations of many microphone pairs has been developed. This algorithm has been implemented in conjunction with a real-time beamformer, but is too computationally intensive at present to produce the desired accuracy and update rate required

for effective beamforming in real-time; this situation typifies the difficulty of applying correlation-based time-delay estimation techniques without a great deal of intelligent pruning (Silverman and Kirtman 1992).

In the next section a frequency-domain delay estimator appropriate for this specific application is described. It is designed to provide high resolution estimates in a single-source environment, to have minimal computational requirements, and to be capable of providing independent delay estimates many times (≈ 70) a second. The delay estimator is evaluated in Section 3 and is shown to be extremely accurate under a wide range of signal conditions. Section 4 presents an application of the delay estimator, in particular, its use as the basis of source location and tracking schemes.

2 Delay Estimator

2.1 Mathematical Development

Consider two microphone receivers in the acoustic-field of a single audio source. Assuming that microphone placement is such that relative signal attenuation between the microphones due to propagation distance and source size and orientation are negligible, the sampled received signals, $r_1(l)$ and $r_2(l)$, may be expressed as:

$$\begin{aligned} r_1(l) &= s(l) + n_1(l) \\ r_2(l) &= s(l-\tau) + n_2(l) \end{aligned} \tag{1}$$

where l is the discrete-time index, $n_1(l)$ and $n_2(l)$ are background noise sources with known statistical characteristics and assumed to be uncorrelated to $s(l)$ and each other, and τ is the relative delay in sample units of the source wavefront between the receivers.

The problem here is to estimate τ from finite-duration sequences of the processes $r_1(l)$ and $r_2(l)$. In typical situations this delay will vary significantly with time, due to the physical movement, (e.g. body and head motion) of the audio source. Measurement consistency is affected by the time-varying nature of the source signal. For instance, a typical speech source may only be considered statistically stationary over a short time frame (≈ 30 ms) and will have periods of signal production interspersed with durations of silence. For these reasons it is advantageous to estimate τ periodically using a small analysis window and to avoid inter-frame averaging in the signal analysis. In what follows, a restriction is imposed that the proposed delay estimator must compute an independent estimate of τ from a single 20-30 ms frame of data.

The DFT coefficients of the N-point, windowed received signals in (1) and their cross-spectrum are given by

$$\begin{aligned} R_1(k) &= W(k) * (S(k) + N_1(k)) \\ R_2(k) &= W(k) * (S(k)e^{-j\omega_k\tau} + N_2(k)) \\ G_{R_1R_2}(k) &= R_1(k)R_2(k)^{'} \end{aligned}$$

where $W(k)$ is the N-point DFT of the analysis window, $k = 0, 1, \ldots, \frac{N}{2}$, $\omega_k = \frac{2\pi k}{N}$, and $*$ and $'$ denote the convolution and complex conjugate operators, respectively. The delay τ now appears as part of the complex phase term and as such, is not restricted to integer values. The phase of the cross-spectrum, may be expressed as

$$\theta_k = arg(G_{R_1R_2}(k)) = \omega_k\tau + \epsilon_k. \tag{2}$$

Here ϵ_k is a random variable that summarizes the contributions of the noise terms and analysis window to the overall phase term at each discrete frequency. Given that ϵ_k is zero-mean for all k (demonstrated in Table 1), the expected value of the phase term, θ_k, is directly proportional to the discrete radian frequency, ω_k, with the constant of proportionality being the signal delay, τ. *i.e.*

$$E(\theta_k) = \omega_k\tau$$

In this sense τ may be interpreted as the slope of the line that "fits" the series of phase terms. Assuming that the ϵ_k terms are uncorrelated (In the case of Gaussian noise sources, this assumption is valid for the wideband speech signals and observation intervals considered here. See (Hodgkiss and Nolte 1976).), the best linear unbiased estimator of τ is given by (Kay 1993):

$$\hat{\tau} = \frac{\sum_{k=1}^{N-1} \omega_k\theta_k[var(\epsilon_k)]^{-1}}{\sum_{k=1}^{N-1} \omega_k^2[var(\epsilon_k)]^{-1}} \tag{3}$$

with

$$var(\hat{\tau}) = \frac{1}{\sum_{k=1}^{N-1} \omega_k^2 var(\epsilon_k)^{-1}} \tag{4}$$

The above analytical expression for calculating $\hat{\tau}$ has several advantages over its time-domain counterpart. It is computationally simple, does not necessitate the use of search methods, and is capable of intra-sample precision (as will be shown in the next section). In addition, if the ϵ_k terms are Gaussian, $\hat{\tau}$ can be shown to be the minimum variance unbiased(MVU) estimator of τ as well (Kay 1993).

2.2 Calculation of Estimator Parameters

In practice, the variance terms required for (3) and (4) are unavailable *a priori* and must be evaluated directly from the data. A means for computing these variances using the magnitude-squared coherence of the spectra derived from overlapping windowed segments is given in (Carter, Knapp, and Nuttall 1973). The conditions required for ϵ_k to be Gaussian and thus $\hat{\tau}$ to be the MVU estimator are stated in (Chan, Hattin, and Plant 1978). A limitation of the method for this application is the long-term averaging required for the coherence estimate. For instance, with a 20 kHz sampling rate and half-overlapping 25.6 ms Hanning windows, the estimate is shown to be equivalent to the maximum-likelihood estimate when averaged over 2 seconds of data. This analysis interval vastly exceeds the independent analysis constraint and is clearly inappropriate for speech signals. A sub-optimal variation of this technique which restricts coherence-estimate analysis to a single 20-30 ms time interval will be considered in the next section.

Given these analysis restrictions, an appropriate alternative is to estimate the error variance independently for each data frame using the following approximation.

$$\hat{var}(\epsilon_k) = \frac{\lambda_{1k}}{\mid R_1(k) \mid^2} + \frac{\lambda_{2k}}{\mid R_2(k) \mid^2} \tag{5}$$

Where the λ_{1k} and λ_{2k} coefficients are derived from the frequency-dependent background noise power at each receiver as follows:

$$\lambda_{1k} = \lim_{J \to \infty} \frac{1}{J} \sum_{j=1}^{J} \mid M_{1j}(k) \mid^2 \tag{6}$$

$$\lambda_{2k} = \lim_{J \to \infty} \frac{1}{J} \sum_{j=1}^{J} \mid M_{2j}(k) \mid^2 \tag{7}$$

with $M_{1j}(k)$ and $M_{2j}(k)$ being the DFT coefficients of individual windowed frames of the background noise sources $n_1(l)$ and $n_2(l)$. The variance estimate may be interpreted as the sum of the approximate inverse S/N ratios at each receiver. Equation (5) was derived assuming relatively large S/N ratios and that the $M_{1j}(k)$ and $M_{2j}(k)$ terms have uniformly random phases.

• Table 1 about here •

Data showing the validity of the assumptions made on the ϵ_k random variables and the accuracy of (5) in estimating the true error variance are presented in Table 1. The results here have been generated with a Gaussian white random source (variance σ_s^2) delayed 1 sample relative to the receivers and then corrupted by uncorrelated additive white Gaussian noise sources (variance σ_n^2). The signals were sampled at 20 kHz, segmented into 200 25.6 ms Hanning windows, and the ϵ_k terms then calculated via (2). The 2 left-hand columns in the table report the sample mean and variance of the ϵ_k terms for each of the S/N conditions. The right-hand column lists the predicted variance of ϵ_k calculated from (5). As the table illustrates, the zero-mean assumption for ϵ_k is appropriate for the entire range of signal conditions. Furthermore, the error variance estimated using (5) accurately models the experimental variance.

S/N (dB)	experimental mean(ϵ_k) $(radians)$	experimental var(ϵ_k) $(radians)^2$	estimated var(ϵ_k) $(radians)^2$
36	.0016	.09	.05
18	-.0052	.52	.36
12	-.0031	.84	.71
0	-.0235	1.77	1.96

Table 1: Experimental mean, variance and estimated variance of ϵ_k as a function of S/N ratio

2.3 Application Considerations

A practical issue that must be considered when applying the proposed estimator is that of phase continuity. The cross-spectrum phase θ_k as evaluated by (2) is modulo 2π whereas the delay estimator (3) requires a phase angle that varies in a continuous linear fashion with the radian frequency. This situation necessitates the use of a "phase-unwrapping" algorithm to remove the 2π discontinuities from the initial θ_k before evaluating $\hat{\tau}$. Several algorithms for this purpose are available from cepstral processing applications, (Tribolet 1977) is typical. An alternative solution to the phase discontinuity problem is given in (Brandstein and Silverman 1993). The "phase-unwrapping" technique used in the following experiments is along the lines of (Tribolet 1977) but less general since the phase difference function is assumed to be linear. As the linear fit in Equation (3) is performed, summing from low frequencies to high frequencies, the intermediate slope estimate at frequency bin k is used to predict the value of θ_{k+1}. The measured value at θ_{k+1} is unwrapped around this predicted value (by adding an integer multiple of 2π to minimize their difference) before adding in the $k+1$ term of the linear-fit sums. A second pass is performed to correct values of θ_k that may been improperly unwrapped due to the variation of the slope estimate over the course of the linear-fit/phase-unwrapping process.

Another practical issue that must addressed is microphone placement. In a near-field setting, such as a room, excessive separation between the microphone receivers may result in significant deviations from the the source model assumptions which have the potential to seriously degrade the quality of the delay estimate. Deviations from the model may be due non-uniform radiation of the source, or unequal signal attenuation and filtering due to the dynamics of the room. Also the short window length employed in the signal processing imposes a practical limit on the maximum delay and therefore the sensor separation. If the delay between 2 channels is an appreciable fraction of the window length one can no longer be confident that there is good correlation between the segments of the source signal captured by the 2 sensors. A feedback mechanism to realign the time sequences by repeating the windowing process with skewed time indexes is not computationally feasible on a frame-by-frame basis. One means of overcoming both these problems is to limit the microphone separation distance. This minimizes near-field radiation effects and restricts the range of potential signal delays. The upper bound for microphone separation distances is dictated by the physical environment and location of signal sources. For the location experiment that follows in Section 4, a microphone separation of 16.5 cm, corresponding to a maximum potential delay of .48 ms, or 9.6 samples at 20kHz, was found to be appropriate for the room enclosure.

3 Delay Estimator Evaluation

3.1 Estimator Experiment 1

Two computer simulations were performed to evaluate the accuracy of the delay estimator described in the previous section. In the first, a single phoneme (the /e/ in 'ketchup') of 20 kHz sampled speech was bandlimited within the range 100 Hz to 5 kHz and isolated with a 25.6 ms Hanning window. Relative delays of 1, 5, and 10 samples were introduced to simulate a second sensor and uncorrelated white Gaussian noise added to both channels. The noise variance was adjusted to create the appropriate S/N ratio. Here S/N ratio is defined as:

$$S/N\,dB = 10\log_{10}\left(\frac{\sum_l [w(l)s(l)]^2}{\sum_l [w(l)n(l)]^2}\right)$$

for a finite length window $w(l)$. A 1024 point FFT was computed for each signal. The delay estimate, $\hat{\tau}$, and the predicted variance of $\hat{\tau}$ were then calculated from (3) and (4), respectively, using the error variance estimates given by (5). Table 2 lists each estimate's sample mean and standard deviation determined from 100 trials at each S/N condition. The values in parentheses represent the averages of the predicted standard deviation. For comparison purposes, the least-squares (LS) delay estimate given in (Chan et al. 1978) was also calculated. The LS estimator involves a weighted line fit of the phase data but differs from the proposed delay estimator in a key respect. The weighting coefficients and phase information required for (3) and (4) are derived via the multiple-window magnitude-square coherence estimation procedure referred to in the preceding section. While the merits of this approach are apparent for long-term statistically and physically stationary signal sources, the expectation is that given the analysis interval time constraint, the LS estimator will be at a disadvantage. A number of scenarios were considered for the partitioning of the 25.6 ms speech segment required for the coherence estimation. The most favorable, which was used to generate the LS estimator results reported in Table 2, incorporated 7 half-overlapping 6.4 ms subwindows.

• Table 2 about here •

While the proposed delay estimator and the LS estimator perform comparably for the small-delay, high-S/N conditions, the experimental results distinctly favor the proposed estimator for the S/N=0 dB case and for the larger sample delays. The LS estimator exhibits a sizeable bias and increased standard deviation at delays of 5 and 10 samples. Part of this effect may be attributed to window misalignment; these delays represent a sizeable fraction of the 6.4 ms subwindows employed by the LS estimator. The proposed estimator with a single 25.6 ms window does not display a marked bias or inflated standard deviation at these larger sample delays. Finally, note that the standard deviation predictor figures from (4) in parentheses accurately model the measured estimator variance for all but the S/N=0 dB case.

sample delay	S/N (dB)	Proposed Estimator		LS Estimator	
		mean	std. dev. (predicted)	mean	std. dev.
1	36	1.00	.017 (.012)	1.00	.014
	24	1.00	.036 (.024)	1.00	.030
	12	1.01	.070 (.048)	1.00	.077
	0	1.00	.162 (.091)	.99	.233
5	36	5.00	.020 (.012)	4.97	.021
	24	5.00	.032 (.024)	4.98	.036
	12	5.01	.074 (.048)	4.99	.077
	0	5.00	.177 (.091)	4.97	.206
10	36	10.00	.017 (.012)	9.93	.028
	24	10.00	.036 (.024)	9.93	.041
	12	10.00	.067 (.048)	9.97	.087
	0	10.03	.182 (.091)	9.99	.265

Table 2: Results of Delay Estimator Experiment 1 (Single phoneme): Sample Mean and Standard Deviation of the Proposed Delay Estimator and the LS Estimator for varying S/N ratios and sample delays. All values are in terms of samples at 20 kHz.

3.2 Estimator Experiment 2

For the second simulation, 100 different frames of randomly-segmented speech representing a wide range of phonemes were prepared under similar conditions to those of the previous experiment. The sample means and standard deviations for the proposed delay estimator and the LS estimator are given in Table 3. In general the variances measured in this experiment are greater than those reported for the single-phoneme experiment. This is most likely due to the varying spectral content of the phonemes used in the second experiment. The /e/ phoneme used in the first experiment is strongly voiced and has very good S/N at the formant frequencies, and thus is well suited to the frequency-dependent weighting used in the delay estimator. In the second experiment the phonemes used cover a broad variety, many of which do not have spectral characteristics quite as favorable for the delay estimation algorithm.

A comparison of the delay estimators' relative performance in this second simulation reveals trends similar, but more pronounced, than those demonstrated in the previous experiment. The larger LS estimator bias is evident even at the high S/N conditions and the variance is larger than that of the proposed estimator's even for the low S/N conditions. The difference in the variances is particularly large for the cases with 5 and 10 sample delays and high S/N. The proposed delay estimator outperforms its counterpart for all the simulation conditions and is relatively insensitive to the differing sample delays although it does begin to show an estimate bias for the S/N=0 dB case.

• Table 3 about here •

The results of these experiments clearly show that the proposed delay estimator has superior performance properties in comparison to the Least-Squares delay estimator presented and is capable of intra-sample precision. For instance, at $S/N \geq 24$ dB, the standard deviation of the estimate is less than .15 samples for all the conditions tested. With regard to computational considerations, the proposed estimator again has a marked advantage over its LS counterpart. For each delay estimate generated in these simulations, the bulk of the proposed delay estimator's computational load is contained in the 2 1024-point FFT's. The remaining elements, such as the calculation of the phases and weights, the phase-unwrapping, and final delay estimation, require computation equal to approximately one-half of a 1024-point FFT. Roughly speaking, the total number of floating point operations required for a single delay estimate is equivalent to that of performing 2.5 1024-point FFT's. The LS estimator used in these experiments needs approximately 3 times this number of operations. A similar correlation-based delay estimator would require a minimum of 3 FFT's to compute the cross-correlation function alone.

sample delay	S/N (dB)	Proposed Estimator mean	Proposed Estimator std. dev.	LS Estimator mean	LS Estimator std. dev.
1	36	1.01	.059	.98	.061
	24	1.00	.144	.96	.205
	12	1.00	.405	.98	.631
	0	.90	.909	.94	1.101
5	36	4.99	.057	4.86	.114
	24	5.01	.147	4.85	.243
	12	4.98	.494	4.77	.641
	0	4.83	1.208	4.74	1.254
10	36	9.99	.061	9.75	.228
	24	9.97	.135	9.70	.462
	12	10.06	.407	9.70	.731
	0	9.90	.999	9.29	1.538

Table 3: Results of Delay Estimator Experiment 2 (100 frames of speech): Sample Mean and Standard Deviation of the Proposed Delay Estimator and the LS Estimator for varying S/N ratios and sample delays. All values are in terms of samples at 20 kHz.

4 Application of the Delay Estimator to Source Location

4.1 Overview

One application of the delay estimator introduced here is as the basis of a source-location algorithm. While the algorithm presented here assumes that the source and sensors are coplanar, the concepts incorporated may be extended to the general case of 3-dimensional source localization with unconstrained placement of microphone pairs (Brandstein 1995). The procedure used obtains a talker's position through a series of triangulation calculations which require knowledge of the signal's relative delay when projecting onto a pair of microphone receivers. The frequency-based delay estimator proposed here is particularly effective in such a scheme for a number of reasons. First, it is highly accurate. The source-location procedure is extremely sensitive to errors in the delay estimate, particularly with distant sources, and therefore precision positional estimates require fine resolution in the delay figure. Second, the computational simplicity and small time window (20-30 ms) associated with the estimator make it possible to generate location estimates at a very high rate. Being able to update a source's location many times a second is essential for effectively tracking and beamforming to a moving source. A time-domain based delay estimator operating at the required resolution and same update rate would require substantially more computing resources to calculate the delay estimates. A third feature of the delay estimator presented above is the variance estimate associated with each delay. The location algorithm incorporates this variance in evaluating the accuracy of a potential source location.

The estimate of the delay between each pair of microphones is used to determine an estimate of the source-bearing relative to the baseline of the microphone pair. The intersections of the source-bearing estimates from different microphone pairs determine the location. Strictly speaking, for a pair of microphones the set of points with a particular inter-microphone delay is a hyperbola, but the error introduced by using a linear bearing estimate is minimal when the microphone baseline is small compared to the distance to the source, and finding the intersections of straight lines is significantly easier than finding the intersections of quadrics.

With an ideal, spherically radiating point source and knowledge of the exact delay values and microphone positions, all the bearing lines intersect at a single point. However, for a more realistic scenario with non-point sources, imperfect delay estimates, and inexact microphone positioning, the source-bearing estimates from more than 2 pairs of sensors will not intersect at a single point. Given these practical conditions, the "optimal" source location will be defined here as the point minimizing a weighted squared-distance function based upon the estimated bearing lines of a set of microphone-pairs. For a particular hypothesized source location, the squared-distance to each bearing line is weighted by a function that approximates the inverse of the variance of that distance. The variance of the distance to the hypothesized location from a particular bearing line, the spatial variance, is derived from the predicted variance of the corresponding delay estimate.

• Figure 1 about here •

Specifically, given 2 receivers and an estimate of the inter-sensor delay, $\hat{\tau}$, the estimated bearing angle of the source relative to the microphone-pair baseline, $\hat{\psi}$, is given by:

$$\hat{\psi} = \arccos\left(\frac{(C/F_s)\cdot\hat{\tau}}{d}\right)$$

where C is the speed of sound, F_s is the sampling rate, $\hat{\tau}$ is the estimated delay in samples, and d is the distance between the microphones. The bearing angle associated with a candidate source

location (x,y) relative to the microphone-pair baseline is denoted by $\phi(x,y)$ and the orthogonal distance from the hypothesized location to the estimated bearing line is given by $\Delta(x,y)$. Figure 1 illustrates this source-receiver geometry.

The orthogonal distance is calculated from:

$$\Delta(x,y) = R(x,y) \cdot \sin(|\hat{\psi} - \phi(x,y)|)$$

Here, $R(x,y)$ is the distance to the candidate location from the microphone-pair midpoint. The variance of $\hat{\psi}$ may be approximated by (Papoulis 1984):

$$var\{\hat{\psi}\} \approx var\{\hat{\tau}\} \cdot \left(\frac{C/F_s}{d \cdot \sin \hat{\psi}} \right)^2$$

and the spatial variance is found from:

$$\begin{aligned} var\{\Delta(x,y)\} &\approx R(x,y)^2 \cdot \cos^2(\hat{\psi} - \phi(x,y)) \cdot var\{\hat{\psi}\} \\ &\approx \left(\frac{R(x,y) \cdot (C/F_s) \cdot \cos(\hat{\psi} - \phi(x,y))}{d \cdot \sin \hat{\psi}} \right)^2 \cdot var\{\hat{\tau}\} \end{aligned}$$

The weighted error function for a given candidate source location (x,y) and a set of M microphone-pairs (indexed by m) is defined to be:

$$E(x,y) = \frac{\sum_{m=1}^{M} W_m(x,y) \cdot \Delta_m(x,y)^2}{\sum_{m=1}^{M} W_m(x,y)} \tag{8}$$

Where the weighting coefficient for microphone pair m, $W_m(x,y)$, is the reciprocal of the estimated spatial variance for that microphone pair. *i.e.*

$$W_m(x,y) = \frac{1}{var\{\Delta_m(x,y)\}}$$

The "optimal" source location estimate, $(\hat{x},\hat{y})$, corresponds to the candidate location found to minimize this error function:

$$(\hat{x},\hat{y}) = \arg\min_{(x,y)} \{E(x,y)\} \tag{9}$$

Since $\Delta(x,y)$ is a non-linear function of the source locations, this error surface is nonconvex and the solution of (9) requires burdensome, and frequently problematic, numerical search methods. For these reasons, it is worthwhile to develop a sub-optimal, closed-form localization method. The procedure outlined below uses the intersections between pairs of estimated bearings to determine a set of points over which to minimize the weighted error function. This restricts the number of evaluations of Equation (8) to a small number.

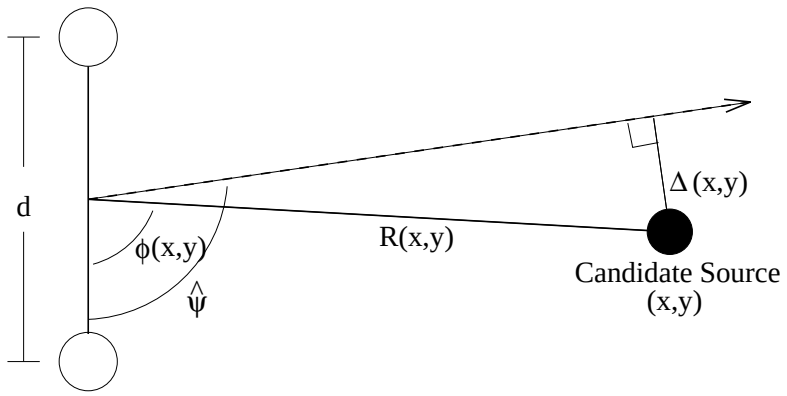

Figure 1: Illustration of the source-receiver geometry. The estimated bearing to the source is shown as a dotted line with angle $\hat{\psi}$ while the bearing associated with a candidate source location is indicated by a solid line with angle $\phi(x, y)$. The microphone-pair separation distance is denoted by d, the candidate source range from the microphone-pair midpoint by $R(x, y)$, and the orthogonal distance from candidate location to estimated bearing line by $\Delta(x, y)$.

4.2 A Source Location Experiment

The experiment was performed within a 3.5 by 4.5 m enclosure. Approximately 70% of the surface area of the enclosure walls is covered with 7.5 cm acoustic foam, the 3 m ceiling is untreated plaster, and the floor is carpeted. The reverberation time within the enclosure is approximately 250 ms. The enclosure is a partially walled-off area contained within an acoustically-untreated workstation lab.

A 2-dimensional "orthogonal array" was used in the experiment. 8 pressure-gradient microphones are horizontally placed at a height of 1.6 m along each of 2 orthogonal walls of the enclosure. The microphones are mounted in a wire mesh offset by 2 cm from the acoustic foam on the wall behind them. Within each 8-microphone sub-array the microphones are uniformly spaced at 16.5 cm intervals. These 2 linear arrays of microphones define the x and y axes and the intersection of the 2 walls determines the origin of the coordinate space.

A speaker was used to play back a recording of the spoken phrase "abc". The speaker has a 5 cm diameter cone and is contained in a sealed enclosure 17 cm on a side and 8 cm deep. The front baffle of the speaker enclosure is covered with sound absorbing foam. The speaker was placed at 20 locations in the room at 70 cm spacings and the recorded phrase simultaneously played back and digitally recorded by the 16 microphones at a 20 kHz sampling rate. The 20 recordings were repeated for each of 3 different speaker orientations: facing the x-axis, facing the y-axis, facing the origin. Figure 2 illustrates the layout of the enclosure and the speaker orientations and positions. For each location, inter-microphone delays were estimated between adjacent receivers using the delay estimator described above with a 25.6 ms window and a half-window shift. The peak recorded signal-to-noise ratio ranged between 15 and 35 dB, varying as a function of speaker distance and orientation relative to the microphone in question. The primary source of background noise in the recording area is computer equipment located both within the experimental enclosure and in the room surrounding the enclosure. The location algorithm used proceeds as follows:

1. Calculate the delay estimate $\hat{\tau}_m$ and corresponding source bearing angle $\hat{\psi}_m$ for each pair of adjacent microphones on the x and y axes. This yields $M = 14$ bearing estimates.

2. For each of the 7 x-axis bearing estimates, find the intersection point with each of the 7 y-axis bearing estimates. This yields K=49 candidate points.

3. Evaluate the weighted error $E(x, y)$ of each of the K intersection points.

4. Sort the points by $E(x, y)$ and choose the $K/2$ points with the lowest error.

5. The final sub-optimal location estimate, $(\tilde{x}, \tilde{y})$, is given by finding the mean coordinate of this low-error subset.

6. A "location-error" estimate is also taken to be the mean of the $K/2$ lowest weighted errors. Frames with a "location-error" exceeding a .04 m^2 detection threshold have no location reported.

• Figure 2 about here •

Figure 2 shows the layout of the array and the recording area along with the speaker positions and the 2σ contour for a Gaussian distribution estimated from each set of location estimates. The speaker was placed at each location by hand, so the "true" locations are only accurate to a few

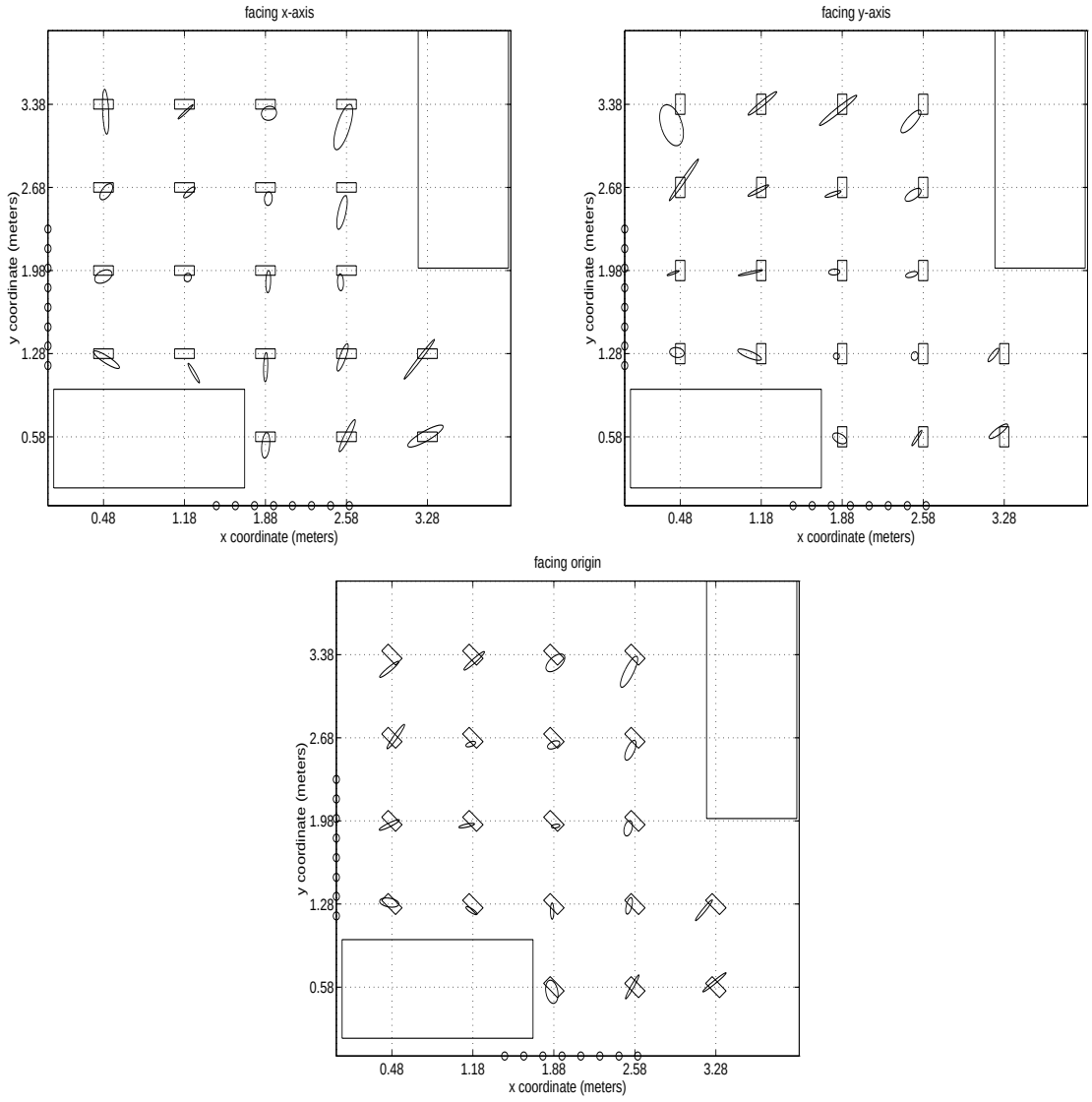

Figure 2: The distribution of position estimates for each location for the 3 speaker orientations: facing the x-axis, facing the origin, and facing the y-axis. Microphone locations are indicated as with an 'o', tables in the room and the speaker enclosure, included to indicate orientation, are shown in outline. The oval is the 2σ contour for a Gaussian distribution based on the position estimates.

centimeters. Because of this and because of the difficulty in attributing a specific "true" point-source location to a 8 by 17 cm speaker enclosure no bias analysis is performed on the computed locations. The distribution of position estimates is calculated using only those analysis frames whose "location-error" falls below the detection threshold and as a consequence the number of valid frames varies for each location with central locations producing more valid frames and peripheral locations fewer. The entire recording contains 69 frames approximately 15 of which are non-speech, yielding an upper limit of 54 valid frames. An average of 30 frames were used to estimate the distribution for each position. For difficult positions (positions not directly in front of either array) as few as 5 valid position estimates were returned while for the easiest positions (positions contained within the dimensions of both linear arrays) the maximum number of valid position estimates was reached. Figure 3 shows the area within the 2σ ellipse for each of the 20 speaker positions. Almost all of the 2σ areas plotted in Figure 3 are within .02 m^2 and the majority are within .01 m^2. This is high precision when one considers that the playback speaker has an aperture diameter of .07 m.

• Figure 3 about here •

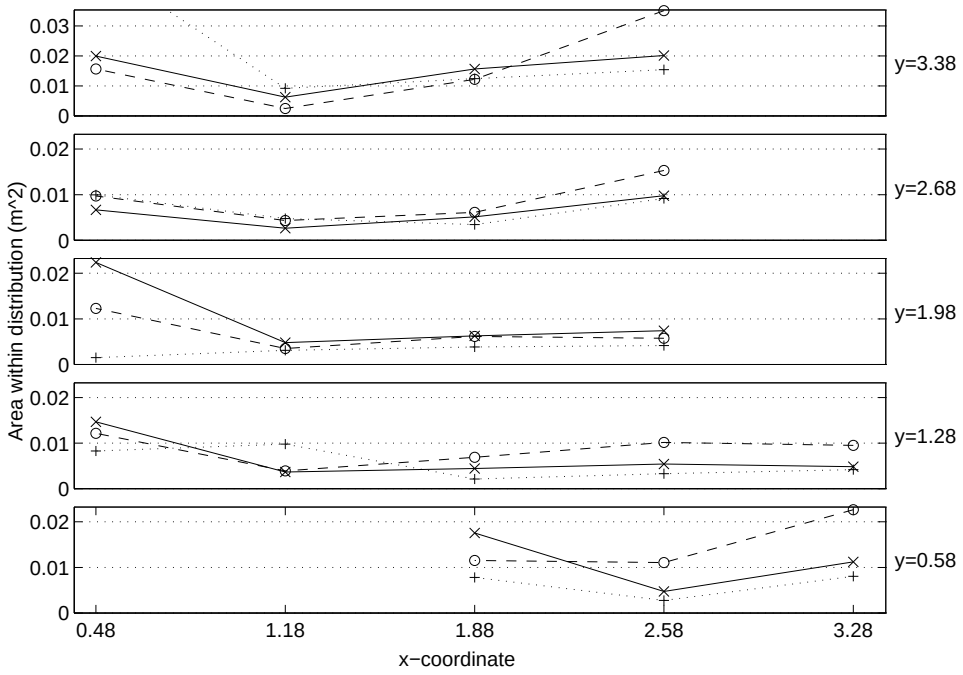

Figure 3: The area of the 2σ ellipse in m^2 for the normal distribution of estimated locations for each position as depicted in Figure 2. The three different marker/line-styles correspond to the 3 different speaker orientations; '+' for the speaker facing the origin, '$\circ$' for the speaker facing the x-axis, and '$\times$' for the speaker facing the y-axis.

4.3 Tracking a Moving Source

The short-time window and high update rate associated with the location algorithm presented above make it appropriate for tracking a moving speech source. With a 25.6 ms half-overlap Hanning window the location algorithm is capable of updating the position estimate 78 times every second. With this brief analysis interval, the source's change of location within the estimation period is insubstantial and has minimal impact on the precision of the calculated delays and derived location.

• Figure 4 about here •

Figure 4 illustrates the ability of the location algorithm to track a moving talker. In this example, a talker spoke the phrase "One-Two-Three-Four" while walking towards the x-axis of the orthogonal array. Source locations were computed exactly as in the previous experiment. Only locations for frames whose "location-error" falls below the detection threshold are plotted. Figure 4a shows the time signal received at a single microphone. The horizontal axis is labeled from zero to 60000 samples, corresponding to three seconds of 20 kHz sampled speech. Figure 4b shows the x and y coordinates of the estimated source position on the same time scale. Note that during periods of silence there is no location reported. The x-coordinate remains constant at approximately 1.9 m, while the y-coordinate decreases gradually throughout the interval, starting at 2.5 m and concluding near 1.3 m. Figure 5 shows these same (x,y) pairs of the location estimates plotted on the grid of the room with the linear fit to these locations as well. The standard deviation of the distance from the estimated locations to the linear fit is .02 m. While there is no mechanism available to accurately determine the exact path traversed by the talker, the smooth, nearly linear path detected by the locator certainly suggests the algorithm is performing accurately.

• Figure 5 about here •

This example does not thoroughly analyze the characteristics of the tracking algorithm, but it does demonstrate its potential. The solution to the problem presented in the introduction, that of beamforming to a moving source, requires the incorporation of an efficient, precise, and frequently updated source locator into the beamforming procedure. The source location system presented here fulfills all these criteria.

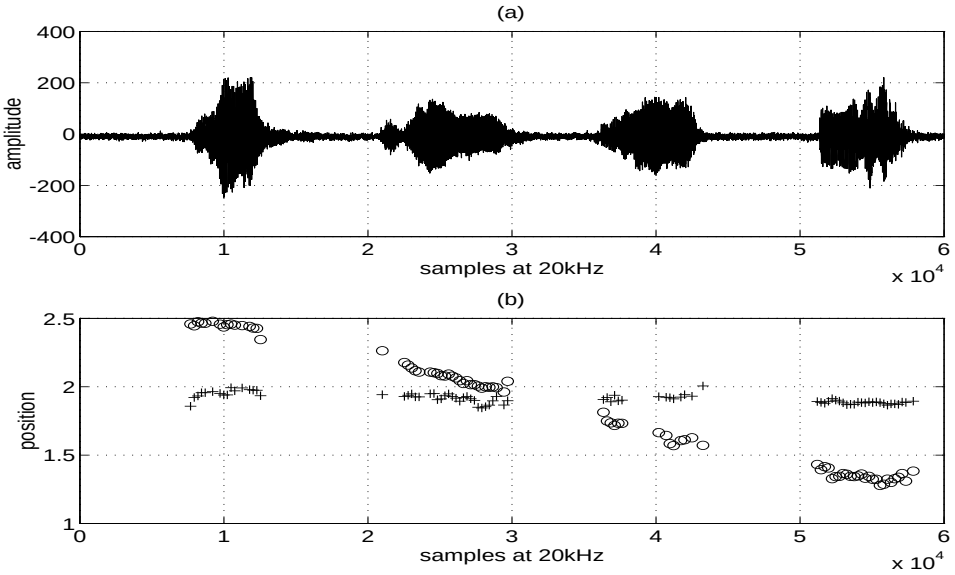

Figure 4: Figure (a) shows the time waveform from a single channel. Figure (b) shows the estimated x-coordinates as +'s and the estimated y-coordinates as o's

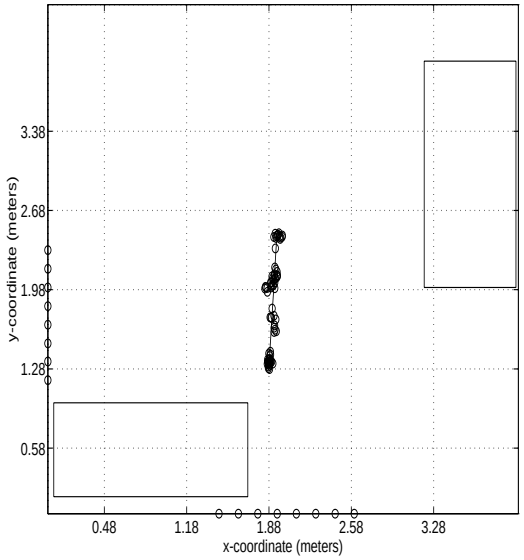

Figure 5: Positions and linear trend derived from a moving talker.

5 Summary

In this paper a frequency-domain based delay estimator, designed specifically for speech signals in a microphone-array environment, has been introduced. It is shown to be capable of obtaining precision delay estimates over a wide range of S/N conditions and is computationally simple enough to make it practical for real-time systems. A location algorithm based upon the delay estimator was then developed. With this algorithm it is possible to localize talker positions to a region only a few centimeters in diameter and to track moving sources. The effectiveness of the delay estimator for these processes demonstrates its potential for other applications. Future work will focus on the development of a general-purpose beamformer which incorporates the source-tracking algorithm.

This work partially funded by NSF Grants MIP-9120843 and MIP-9314625. We gratefully thank the reviewers for their comments which we found to be insightful and useful for improving the quality of the manuscript.

References

Alvarado, V. M. (1990, May). *Talker Localization and Optimal Placement of Microphones for a Linear MIcrophone Array using Stochastic Region Contraction.* Ph. D. thesis, Brown University, Providence, RI.

Brandstein, M. S. (1995, May). *A Framework for Speech Source Localization Using Sensor Arrays.* Ph. D. thesis, Brown University, Providence, RI.

Brandstein, M. S. and H. F. Silverman (1993, March). A new time-delay estimator for finding source locations using a microphone array. LEMS Technical Report 116, LEMS, Division of Engineering, Brown University, Providence, RI 02912.

Carter, G. C., C. H. Knapp, and A. H. Nuttall (1973, August). Estimation of the magnitude-squared coherence function via overlapped fast fourier transform processing. *IEEE Transactions Audio and Electroacoustics AU-21*(4), 337–344.

Chan, Y. T., R. V. Hattin, and J. B. Plant (1978). The least squares estimation of time delay and its use in signal detection. *IEEE Transactions on Acoustics, Speech, and Signal Processing 26*, 217–222.

Che, C., Q. Lin, J. Pearson, B. deVries, and J. Flanagan (1994, March 8-11). Microphone arrays and neural networks for robust speech recognition. In *Proceedings of the Human Language Technology Workshop*, Plainsboro, NJ, pp. 342–347. ARPA-Software & Intelligent Systems Technology Office.

Flanagan, J. L. (1985, March). Bandwidth design for speech-seeking microphone arrays. In *Proceedings of ICASSP85*, Tampa, FL, pp. 732–735.

Flanagan, J. L., J. D. Johnson, R. Zahn, and G. W. Elko (1985, November). Computer-steered microphone arrays for sound transduction in large rooms. *J. Acoust. Soc. Amer. 78*(5), 1508–1518.

Flanagan, J. L. and H. F. Silverman (1992, October). Material for international workshop on microphone array systems: Theory and practice. Technical report, Division of Engineering, Brown University, Providence, RI 02912.

Greenberg, J. E. and P. M. Zurek (1992, March). Evaluation of an adaptive beamforming method for hearing aids. *J. Acoust. Soc. Amer. 91*, 1662–1676.

Grenier, Y. (1992, April). A microphone array for car environments. In *Proceedings of ICASSP92*, San Francisco, CA, pp. I–305 – I–309.

Hassab, J. C. (1989). *Underwater Signal and Data Processing.* CRC Press, Inc.

Hodgkiss, W. S. and L. W. Nolte (1976, March). Covariance between fourier coefficients representing the time waveforms observed from an array of sensors. *J. Acoust. Soc. Am. 59*, 582–590.

Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing.* Prentice Hall, Inc.

Kellerman, W. (1991, May). A self-steering digital microphone array. In *Proceedings of ICASSP91*, Toronto, CA, pp. 3581–3584.

Knapp, C. H. and G. C. Carter (1976, August). The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Process. ASSP-24*(4), 320–327.

Lin, Q., E. Jan, and J. Flanagan (1994, October). Microphone arrays and speaker identification. *IEEE Transactions on Speech and Audio Processing 2*(4), 622–629.

Oh, S., V. Viswanathan, and P. Papamichalis (1992, April). Hands-free voice communication in an automobile with a microphone array. In *Proceedings of ICASSP92*, San Francisco, CA, pp. I–281 – I–284.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill.

Silverman, H. F. (1987, December). Some analysis of microphone arrays for speech data acquisition. *IEEE Trans. Acoust. Speech Signal Process. ASSP-35*(2), 1699–1712.

Silverman, H. F. and S. E. Kirtman (1992, April). A two-stage algorithm for determining talker location from linear microphone-array data. *Computer, Speech, and Language 6*(2), 129–152.

Tribolet, J. M. (1977). A new phase unwrapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing 25*, 170–177.